

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

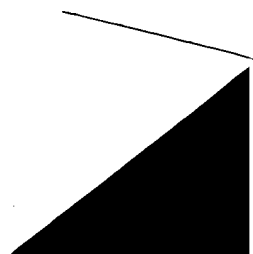
Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**





## PATENT ABSTRACTS OF JAPAN

(11) Publication number: 07114519 A

(43) Date of publication of application: 02 . 05 . 95

(51) Int. Cl. G06F 15/16  
G06F 9/06

(21) Application number: 05260842

(22) Date of filing: 19 . 10 . 93

(71) Applicant: HITACHI LTD

(72) Inventor: KONO KATSUMI  
MORI KINJI  
OSHIMA KELJI  
NAKAMURA TOMOAKI

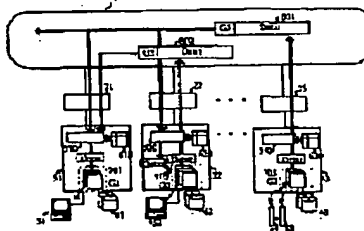
(54) METHOD FOR MANAGING DIFFERENT KIND OF management, etc.  
OPERATION

COPYRIGHT: (C)1995,JPO

(57) Abstract:

**PURPOSE:** To manage different kinds of operation and information by their attributes when those kinds of operation and information are present in one decentralized system by grouping program modules by operation discriminations and performing autonomous execution management and sent and received data management, group by group.

**CONSTITUTION:** Program modules 701 and 702 are discriminated with the same identifier G1 and a program module 703 is discriminated with an identifier G2. When a program module which belongs to the program module group 703 outputs data, the group identifier G1 is added to the data, which are outputted to a common transmission medium 1. Similarly, when a program module in the program module group 702 outputs data, the group identifier G2 is added to the data, which are outputted to the common transmission medium 1. Those messages are sent to other processors through the common transmission medium 1 and used for execution



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-114519

(43) 公開日 平成7年(1995)5月2日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 15/16  
9/06

識別記号

4 6 0 T 7429-5L  
4 1 0 A 9367-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数23 O L (全 12 頁)

(21) 出願番号

特願平5-260642

(22) 出願日

平成5年(1993)10月19日

(71) 出願人

000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者

河野 克己

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者

森 欣司

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者

大島 啓司

茨城県日立市大みか町五丁目2番1号 株

式会社日立製作所大みか工場内

(74) 代理人

弁理士 小川 勝男

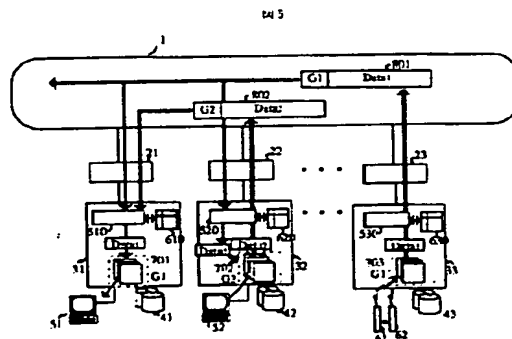
最終頁に続く

(54) 【発明の名称】 異種業務管理方法

(57) 【要約】

【目的】 一つの分散システム内に異種の業務が存在したり、その異種業務により扱われる異種の情報が存在する場合に、それらの業務や情報をその属性ごとに管理することが可能にする。

【構成】 プログラムモジュールを業務種別ごとにグループ化し各グループ毎での自律的な実行管理ならびに送受信データ管理を行ない、特定のグループの実行時間を保証したり、あるいは自らのグループへの他のグループからのデータの受信を抑止したりすることを、各々のグループが自らの業務グループの特性から判断して行ない、該業務が保証すべき特性を保証させる。



## 【特許請求の範囲】

【請求項1】複数の処理装置と、該処理装置間を結ぶ共通伝送媒体ならびに処理装置内での複数のプログラムモジュール間のデータ授受を行なう共通伝送媒体と、前記処理装置間での前記伝送媒体を介したデータの授受を可能とする各処理装置ごとの該伝送媒体との伝送制御を行う伝送制御装置とを備え、複数の仕事に関わる複数の一連の処理を実行する分散計算機システムの異種業務管理方法において、前記処理装置のそれぞれに保持されたプログラムモジュールを該プログラムモジュールが実行する処理が対応する仕事、あるいは、仕事の種類ごとにグループ分けし、各プログラムモジュールの実行管理を前記プログラムモジュールが属するグループごとに行うことを特徴とする異種業務管理方法。

【請求項2】請求項1記載の異種業務管理方法において、前記グループの各々毎に、属するプログラムモジュールによる一連の処理時間を監視し、特定のグループの処理経過時間が所定の時間を超えた場合には他のグループのプログラムモジュールの処理の実行を中断させる、ことを特徴とする異種業務管理方法。

【請求項3】請求項1記載の異種業務管理方法において、各プログラムモジュールの一部のみを仕事の種類ごとにグループ分けし、前記グループのいずれかに属するプログラムモジュールと前記グループのいずれにも属さないプログラムモジュールを混在させて管理することを特徴とする異種業務管理方法。

【請求項4】請求項1記載の異種業務管理方法において、一つのグループを一つの処理装置に対応させ、一つの処理装置内では一つのグループのプログラムモジュールのみを稼働させることを特徴とする異種業務管理方法。

【請求項5】請求項1記載の異種業務管理方法において、一つの処理装置内にそれぞれ異なるグループに属する複数のプログラムモジュールを存在させることを特徴とする異種業務管理方法。

【請求項6】請求項1記載の異種業務管理方法において、各プログラムモジュールをどのグループに属させるかを事前に設定し、該設定情報に従ってプログラムの実行管理を行なうことを特徴とする異種業務管理方法。

【請求項7】請求項1記載の異種業務管理方法において、各プログラムモジュールをどのグループに属させるかを該プログラムモジュールの稼働状態から判断して設定し、該設定情報に従ってプログラムの実行管理を行なうことを特徴とする異種業務管理方法。

【請求項8】請求項1記載の異種業務管理方法において、各グループごとに該グループを識別するグループ識別子を定め、グループ内のプログラムモジュールがその処理結果として出力するデータに該グループ識別子を付加して共通伝送媒体に出力し、複数のプログラムモジュール間でのデータの受渡しを、共通伝送媒体からデータ

を受け取る際に自らのグループの識別子と同一の識別子が付加されたデータのみを受信することを特徴とする異種業務管理方法。

【請求項9】請求項8記載の異種業務管理方法において、各グループ内のプログラムモジュールと他グループ内のプログラムモジュールとの間のデータの授受をプログラムモジュールがその処理結果として出力するデータにグループ識別子に加えて、該データの内容を示す内容識別子も付加して行い、各グループのプログラムモジュールは該データのグループ識別子と内容識別子とから該データを受けとって良いかを判断することを特徴とする異種業務管理方法。

【請求項10】請求項8記載の異種業務管理方法において、各グループ内のプログラムモジュールとグループに属さないプログラムモジュール間のデータの授受を該データの内容を示す内容識別子を用いて行なうことを特徴とする異種業務管理方法。

【請求項11】請求項8記載の異種業務管理方法において、一つのグループ内のプログラムモジュール間のデータの授受を該データの内容を示す内容識別子を用いて行なうことを特徴とする異種業務管理方法。

【請求項12】請求項1記載の異種業務管理方法において、各プログラムモジュールがアクセスするファイルやテーブルあるいは入出力装置をグループ毎に分離し、あるグループに属するプログラムモジュールは他のグループのファイルやテーブルあるいは入出力装置にアクセスできないようにすることを特徴とする異種業務管理方法。

【請求項13】請求項1または3記載の異種業務管理方法において、一つの処理装置内の各グループ内の複数のプログラムモジュールは同時には実行させないことを特徴とする異種業務管理方法。

【請求項14】請求項8記載の異種業務管理方法において、一つのグループのプログラムモジュールが受信した一つのデータに関わる該グループ内での一連の処理時間が一定時間を越えた場合に該一連の処理をその時点で終了させることを特徴とする異種業務管理方法。

【請求項15】請求項8記載の異種業務管理方法において、一つのグループのプログラムモジュールが受信した一つのデータに関わる該グループ内での一連の処理時間が一定時間を越えた場合に、異常通知を行うことを特徴とする異種業務管理方法。

【請求項16】請求項12記載の異種業務管理方法において、自らのグループのファイルやテーブルへのアクセスを、特定のグループからのみ許可することを特徴とする異種業務管理方法。

【請求項17】請求項12記載の異種業務管理方法において、自らのグループのファイルやテーブルの更新や参照を他のプログラムモジュールからの要求に応じて行なうファイル管理プログラムモジュールを自グループ内のプ

ログラムモジュールとして設けることを特徴とする異種業務管理方法。

【請求項18】請求項16記載の異種業務管理方法において、自内のファイル管理プログラムモジュールに対する要求が他グループからの要求がどうかを判断し、他グループからの要求である場合に該ファイル管理プログラムの実行時間を監視し、該実行時間が所定の時間を越えた場合には前記ファイル管理プログラムの処理をその時点で終了させることを特徴とする異種業務管理方法。

【請求項19】請求項16記載の異種業務管理方法において、前記ファイル管理プログラムモジュールへの要求をプログラムモジュール間で授受するデータを用いて行うことを特徴とする異種業務管理方法。

【請求項20】請求項7記載の異種業務管理方法において、各プログラムモジュールをどのグループに属させるかを、該プログラムモジュールの処理時間から判断してある範囲に入るプログラムモジュールを同一グループとして設定し、該設定情報に従ってプログラムの実行管理を行なうことを特徴とする異種業務管理方法。

【請求項21】請求項7記載の異種業務管理方法において、各プログラムモジュールをどのグループに属させるかを該プログラムモジュールの稼働状態から判断して設定し、該設定情報に従ってプログラムの実行管理を行ない、前記稼働状態の監視を繰り返し、前記稼働状態が変化した場合には前記設定情報を修正することを特徴とする異種業務管理方法。

【請求項22】請求項7または20記載の異種業務管理方法において、各プログラムモジュールをどのグループに属させるかを、該プログラムモジュールの処理時間から判断してある範囲に入るプログラムモジュールを同一グループとして設定し、該設定情報に従ってプログラムの実行管理を行なうことを特徴とする異種業務管理方法。

【請求項23】請求項7または20記載の異種業務管理方法において、各プログラムモジュールをどのグループに属させるかを、該プログラムモジュールが入力するデータ或いは出力するデータの種類の種類から判断して同一種類のデータを入力あるいは出力するプログラムモジュールを同一グループとして設定し該設定情報に従ってプログラムの実行管理を行なうことを特徴とする異種業務管理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、複数種の業務に関する複数の処理の各々を複数の処理装置で分散して行なう分散計算機システムに係り、とくに交通、FAなど制御や情報処理など性質の異なる業務に関する処理を混在して行なう分散計算機システムの異種業務管理方法に関する。

【0002】

【従来の技術】従来、ある種の業務に係る一連の処理を複数の処理装置で分散して行なうような分散計算機システムにおいて、処理装置として種類の異なる計算機を用いてシステムを構築する際に、これら種類の異なる計算機の間でのデータの授受を単に透過にする、即ち、データをどの計算機からでも送受信できるようにすることを可能とする技術が知られている。

【0003】

【発明が解決しようとする課題】しかし、従来より実現されている分散計算機システムは、種類の異なる業務をシステムとして互いに連携を取って各々の業務の特性を生かして稼働させる（業務の特徴に応じて例えばリアルタイム性を保証したりする）といった点については、何等考慮がなされていなかった。近年、特に、交通、FA等の分野では、顧客自身のニーズの多様化に伴い一つのシステムの中で異種の業務が混在せざるを得ない状況となっており、これら異種業務間での情報の分離と統合が大きな技術課題となってきた。

【0004】高度な情報サービスを行なうためには、異種の業務を一つの計算機システムの中で混在させてそれぞれの特徴を活かした処理を行なわせる必要がある。このためにはシステム内の各異種業務の処理を行なう実体であるプログラムモジュールと異種の業務が生成する情報の属性を如何に管理し情報を統合させたり分離させたりするかが課題となる。

【0005】

【課題を解決するための手段】本発明による異種業務管理方法では、上述した技術課題を解決するために、プログラムモジュールを業務種別ごとにグループ化し各グループ毎での自律的な実行管理ならびに送受信データ管理を行ない、特定のグループの実行時間を保証したり、あるいは自らのグループへの他のグループからのデータの受信を抑止したりすることを、各々のグループが自らの業務グループの特性から判断して行ない、該業務が保証すべき特性を保証させる。

【0006】

【作用】上述したような方法により、異種の業務が混在する分散システムにおける情報の流れを、その属性ごとに自律して管理することが可能となり、ある属性の情報が特定のグループに流れ込むのを抑止したり、複数の属性の情報を一つに統合することができる。この結果、異種業務間での属性の保護や連携が可能となる。

【0007】

【実施例】以下本発明の実施例を図面を用いて詳細に説明する。

【0008】図2は本発明が適用される分散計算機システムの構成の一例を示すブロック図である。図2において、31、32、及び33は処理装置。1は処理装置31、32、33の間を結ぶ共通伝送媒体。21、22、23は共通伝送媒体1と処理装置31、32、33との

間でのデータの授受をそれぞれ制御する伝送制御装置。

41、42、43は各処理装置31、32、33での処理に必要なデータやプログラムを格納する記憶媒体である。又、51は人（オペレータ）がデータの入出力を行なう端末装置であり、61、62はセンサなど外部からのイベント情報を制御する入出力装置である。

【0009】本実施例の分散計算機システムでは、異なる種類の業務プログラムが互いに共通伝送媒体を介してデータを授受し、連携を取り処理を行なうものとする。ここで、異なる種類の業務とは、例えば、端末51から人が入力したコマンド（要求データ）に応じ、記憶媒体41、42或いは43の中にあるデータを検索するようなオンデマンド業務と、センサ61や62から得られる時々刻々のデータに応じて制御処理を行なうリアルタイム業務、など性質の異なる業務を指すものとする。

【0010】図3は、本実施例において、プログラムモジュール間で授受されるメッセージ300のフォーマットを表したものである。図3において、301のGIDはメッセージ300に含まれるデータ302を出力したプログラムモジュールが属するグループ（後述）を示すグループ識別子である。304は、いわゆるプロトコルヘッダ（PH）と呼ばれるものであり、また303のはトレイラ（CRC）と呼ばれるものである。CRC303及びPH304は各々メッセージ300全体を処理装置間でやり取りするための手順（プロトコル）と誤り処理に関する管理情報のうち、本方式に固有の部分を除く部分に関するものである。PH304には、例えば、従来の伝送方式で用いられる送り先アドレスなどが、またCRC303には、例えば、CRCチェック方式と呼ばれる誤り処理方式のコードデータなどが書かれているものとする。

【0011】本実施例は、処理装置間で授受されるメッセージを該メッセージを使用する業務の種類毎に管理するものであるが、下位層の伝送手段としてはこのように従来の伝送方式を用いても何ら問題ない。

【0012】図4は、メッセージ300内のPH304として、特に内容コードCC400を用いたメッセージのフォーマットを示している。CC400は、データ302の内容を表すコードである。以下、本実施例の説明では共通伝送媒体を介したデータの伝送を内容コード400を用いた処理装置間の通信によって行う例を挙げるが、もちろん他の通信方式、例えば従来からある1対1通信方式などを用いてもよい。ただし、本実施例では、以下に説明するように、グループ化された各プログラムモジュールの実行時間の監視や送受信するデータの管理を各グループごとに、集中管理機能なしで（すなわち分散システム上での他のグループの存在を意識することなく）行うことを特徴としている。つまり、本実施例のように、内容コードを用いた通信方式を用いることにより、システム全体としてより柔軟で拡張性保守性に富ま

せることができるという別の効果を引きだすことができる。

【0013】図5は、プログラムモジュール間でデータの授受を行なう場合のデータの流れの一例示すシステムフロー図である。図5において、701、702、703は各処理装置内のプログラムモジュールグループであり、G1、G2は各グループの識別名を示している。もちろん後述するようにグループは一つの処理装置の中に複数存在しても良い。510、520、530は各処理装置内でプログラムモジュールグループの管理を行う管理プログラムであり、所謂OS（オペレーティングシステム）に相当するソフトウェアである。610、620、630は管理プログラム510、520、530が各々用いる管理テーブル（詳細後述）であり、本発明方式を実現する上で必要な情報が書き込まれているものとする。801と802は各々G1とG2というグループ内のプログラムモジュールから出力されたデータ（各々Data1とData2）の共通伝送媒体上を流れるメッセージを示すものである。矢印はシステム内でのメッセージ（データ）の流れを示している。但し、メッセージ801、802は、図3におけるPH304とCRC303の部分は省略して示している。

【0014】図5ではプログラムモジュールグループ701と703が同一の識別子G1で識別されるグループとなっている。また、プログラムモジュールグループ702は、識別子G2により識別される。

【0015】いま、プログラムモジュールグループ703に属するある一つのプログラムモジュールがデータを出力すると、そのデータにグループ識別子G1が付加されて、共通伝送媒体1に出力される。同様にプログラムモジュールグループ702内のある一つのプログラムモジュールがデータを出力すると、そのデータにグループ識別子G2が付加されて共通伝送媒体1に出力される。これらのメッセージは共通伝送媒体1を介して他の処理装置へ送られる。送り方は従来の伝送プロトコルを利用して、例えば送り先処理装置（図5の例では処理装置31）のアドレスを指定したり、あるいは内容コードを用いる伝送方式を用いる。内容コードを用いる場合には、データを送信する際に、そのデータに対応する内容コードを付加し、共通伝送媒体に対して報知される。この場合各処理装置は、内容コードを指定するのみで共通伝送媒体を流れてくる該当するデータを受け取ることができる。図5に示す例では、処理装置31は、グループ識別子G1、G2のメッセージを受けとり、処理装置32では、グループ識別子G1のメッセージを受け取っている。

【0016】本実施例では、処理装置31、32の管理プログラム510、520はそれぞれ異なった管理方式により受信するメッセージを選択している。

【0017】まず、処理装置31内の管理プログラム5

10は、メッセージに付加されているグループ識別子を  
チェックし、自らが管理しているプログラムモジュール  
グループの識別子と比較する。そして比較の結果、両者  
が一致した場合に、そのメッセージを自らが管理してい  
るプログラムモジュールグループ内の適切なプログラム  
モジュールに送る。従って処理装置32から送られてき  
たメッセージ802は管理プログラム510により廃棄  
され処理装置33から送られてきたメッセージのみがプ  
ログラムモジュールグループ701に送られることとな  
る。この方式では、プログラムモジュールグループ70  
1内のプログラムモジュールがある内容コードを持つデ  
ータの受信を要求したとしても、グループ識別子が異な  
るプログラムモジュールグループからのメッセージは受  
信できない。この方式により、例えばリアルタイム性が  
要求される業務に関わるプログラムモジュール群を一つ  
のグループにまとめておき、他の種類の業務からのデー  
タによってリアルタイム性を損なう余計な処理が行われ  
るのを防ぐことができる。

【0018】次に、処理装置32では、異なるプログラ  
ムモジュールグループからのメッセージも取り込んで自  
らが管理するプログラムモジュールグループ内のプログラ  
ムモジュールに渡している。この場合は、管理テー  
ブル620上に受信メッセージを自らが管理するグループ  
のプログラムモジュールに渡してもよい情報が送られて  
くるプログラムモジュールグループのグループ識別子と  
して自らが管理するプログラムモジュールグループのほ  
かに他のグループの識別子も設定しておき管理プログラ  
ムによって受け渡しの要否の判定が行われる。あるいは  
また、受け渡しをしてはいけないグループの識別子のみ  
を管理テーブルに設定して、そのグループ以外からのメ  
ッセージは全て受け渡す方式としてもよい。

【0019】各処理装置において、これらの管理方式の  
うちいずれの管理方式を採用かは、プログラムモジュ  
ールグループごとにそのグループ識別子を考慮して決めら  
れ、集中的に一括して決める必要はない。

【0020】図6は一つの処理装置の中に複数のグル  
ープが存在する場合の実施例である。図5で説明した実施  
例は、処理装置間でメッセージ（データ）を授受するも  
のであるが、図6に示すように、メッセージの授受を一  
つの処理装置内のプログラムモジュールグループ73  
0、740の間で行ってもよい。更に、同一のプログラ  
ムモジュールグループ内のプログラムモジュール間で、  
データの授受を行ってもよい。この場合、管理プログラ  
ムでは処理装置外から受信したメッセージも処理装置内  
でのメッセージも同一の手順によって処理できる。

【0021】なお、処理装置内には、グループ管理を行  
なわず、通常のメッセージ送受信のみを行なうグル  
ープ外のプログラムモジュールが存在しても良い。これらグ  
ループ外のプログラムモジュールがグループ内のプログラ  
ムモジュールとの間でメッセージのやり取りを行なう

場合には、メッセージ中のグループ識別子が空欄（設定  
しない）の状態で行なえばよい。上記図5の説明で述べ  
たグループ識別子の比較チェック等を変える必要はな  
い。

【0022】図7は、ある一つの業務にかかわる一連の  
処理の時間管理を説明するための処理装置内のプログラ  
ムモジュール間の関係を表すシステムフロー図である。

【0023】図7において、7101、7102、72  
01はデータの流れを示し、7202はプログラムモ  
ジュール間での起動関係（コントロールの流れ）を示して  
いる。グループ701内のプログラムモジュール701  
1がデータを受けて処理を行ない、その処理結果のデー  
タをプログラムモジュール7012に渡している。プロ  
グラムモジュール7012は自らの処理を行ないプロ  
グラムモジュール7013に起動をかけている。プログラ  
ムモジュール7012に起動されたプログラムモジュ  
ール7013は、更に他のプログラムモジュールヘデー  
タを渡す。このように、データ入力7101に起因する一  
連の処理が行なわれ最後にプログラムモジュール701  
4が、処理結果のデータを7102のように送り出して  
いる。こうした一連の処理に要する時間は各グループ毎  
に管理プログラムによって管理される。管理プログラム  
は所定のグループに関して、一連の処理の時間が所定の  
時間を越えた場合には、他のグループの処理の続行を制  
限する。

【0024】図8は各処理装置内のソフトウェアの内部  
構造を示すブロック図である。

【0025】図8において、510は自処理装置内のプ  
ログラムモジュールを管理する管理プログラムである。  
この管理プログラム510は、各々の管理テーブル61  
0、611を用いて図1に示す処理フローに従って、自  
処理装置内のプログラムモジュールの実行、ならびにこ  
れらプログラムモジュールのデータ入出力を管理を行な  
う。1100、1200は、各々共通伝送媒体を解して  
メッセージを送受信する際に用いる受信バッファ、なら  
びに送信バッファである。共通伝送媒体を流れるメッ  
セージは、通信ドライバプログラム81によって受信バッ  
ファ1100に取り込まれる。受信バッファ1100内  
の1005、1006は、それぞれ取り込まれたメッ  
セージを表している。また、管理プログラム510から出  
力されたメッセージ1004は、送信バッファ1200  
に格納され、通信ドライバ81によって共通伝送媒体へ  
送信される。

【0026】管理プログラム510は受信バッファ11  
00からメッセージを取り込むと、取り込んだメッ  
セージのヘッダ部に設定されている情報と管理テーブル61  
0の送受信管理情報部に設定されている情報とを比較し  
て、そのメッセージ内のデータをどのプログラムモジュ  
ールへ渡せばよいかを判定する。図9に、管理テーブル  
610の詳細な構成図を示す。本実施例では、メッセー

ジ送受信のプロトコル(手順)として内容コードを用いたものである。

【0027】更に、管理プログラム510は、取り込んだメッセージ内のデータをどのプログラムモジュールへ渡せばよいかを判定した後、管理テーブル610に設定されているプログラムモジュールの属するグループのグループ識別子(あるいは受信可能な相手のグループ識別子)とそのデータに付加されているグループ識別子とを比較し、そのデータをプログラムモジュールに渡してよいかどうかを判定する。判定の方式は図5により説明したごとく各グループの性質に対応して行う。

【0028】本実施例において、図7に示した各グループの処理時間を管理する場合には、管理テーブル611を用いる。この管理は、一つのプログラムモジュールグループにデータを渡してから後、そのグループからの出力データが管理プログラム510に渡されるまでの経過時間を監視することによって行う。管理プログラム510と各グループ内のプログラムモジュール71~74との間のデータの受け渡しは、両者の間にある中間バッファ101ならびに102を用いて行う。管理プログラム510から各グループ内のプログラムモジュールへ渡されるデータ1002は、まず管理プログラム510が中間バッファ101に格納する。この時、管理プログラム510は、管理テーブル611上の処理時間タイマーを0からスタートさせる。以後、このグループからの出力データ1003が中間バッファ102を介して管理プログラム510へ渡されるまで処理時間の監視を続ける。処理時間が管理テーブル上に示されるそのグループの処理の許容時間を越えた場合には、対応する排他グループ識別子を持つグループの実行を抑止する。許容時間内にデータが出力された場合には611の処理時間をリセットする。抑止の方法としては、処理時間が許容時間内に収まるまで排他グループへのデータを渡さないようにするか、あるいは、排他グループ内のプログラムモジュールの実行そのものを一時停止させることによって行う。排他グループのプログラムモジュールの実行の一時停止は、従来のOS(オペレーティングシステム)の持つ機能を管理プログラム510が利用することによって行うことが出来る。

【0029】以上説明した管理プログラム510での時間管理の処理フローを図13に示す。

【0030】図13に示す処理は、図1に示す管理方式により更新される管理テーブル611を定期的にチェックして、各グループの一連の処理時間が許容時間を越えていないかどうかをチェックするものである。ここで一連の処理時間とは共通伝送媒体からデータを受け取ってプログラムモジュールが起動されてから、そのグループ内でデータの授受を含む処理が行われ、グループ内では使わないデータを共通伝送媒体に送信するまでの時間である。

【0031】まず、管理テーブル611を検索して各グループの処理時間を参照する(13001)。処理時間が許容時間を越えている( $t_i > T_i$ となっている)グループがある場合には(13002)、排他グループの抑止フラグを立ててプログラムの実行を抑止して(13003)、一定時間ウエイトする(13004)。処理時間が許容時間を越えているグループがない場合には、排他グループの抑止フラグを見て実行が抑止されていれば抑止を解除して一定時間ウエイトする(13005)。

【0032】図10、図11は、プログラムモジュールのファイル管理に関する別の実施例である。本実施例の方式では、図7に示すように主記憶61やディスク41上のファイルを各グループ内でクローズさせ、他グループに属するファイルには直接アクセスできないようにファイル管理を行なう。この管理は例えば、ファイルアクセスマクロ内でコールしたプログラムモジュールがどのグループに属するかをチェックして、他グループのプログラムモジュールの場合には異常リターンさせるなどの方法により実現できる。本方式では図8に示すように、グループに属するファイルへのアクセスは共通伝送媒体を介したメッセージを用いて行なう。即ち、ファイルへのアクセス要求であるメッセージ3001を送信し、このファイルを有するグループでこのファイルを直接管理しているプログラムモジュール4012が要求に応じたアクセスを行なってその結果をメッセージ3002にして共通伝送媒体に送出する。この場合も前述したように、メッセージの属性チェックを行ない属性の異なるグループからのアクセスを禁止(抑止)することができ

る。

【0033】図12は、各処理装置内のハードウェア構成を表すブロック図である。

【0034】図12において、3101は、伝送制御装置21との信号の授受を制御するLANインターフェースユニット、3102は各種プログラムやデータあるいはテーブルなどを格納するメモリユニット、3103は各種プログラムを各種データやテーブルを用いて実行処理するプロセッサユニット、3104、3105は各々端末装置51と外部記憶装置(ディスクなど)41とのインターフェースユニット。3106は内部クロック(タイマー)を管理するタイマ管理ユニット。これら各ユニットは、処理装置内のバス3107を介して接続されている。

【0035】図1は、各処理装置内の管理プログラムが実行する処理の流れを示す処理フロー図である。管理プログラムは1100で一度起動し処理を開始すると、送受信バッファや自処理装置内のアプリケーションプログラムからの出力データの有無を常時監視して動き続ける。管理プログラムは、処理装置の立ち上げ時や、通信ドライバプログラムからの受信通知があった際に、最初

50 に起動される。以下、管理プログラムの処理手順を示



す。

【0036】管理プログラムはまず受信バッファを見て、受信バッファに共通伝送媒体からの受信メッセージが有るかどうかをチェックする(1101)。受信バッファにメッセージがある場合には、受信バッファの先頭位置からメッセージを取り出す(1102)。そして、管理テーブルの受信CCエリアを検索してメッセージに付加されている内容コードと合致するものを探し、合致したCCのアプリケーションプログラムモジュールを特定する(1103)。更にメッセージに付加されている情報属性識別子と特定したアプリケーションプログラムモジュールに関する管理プログラム上の受信情報属性識別子とを比較し、両者が合致するかどうかをチェックする(1104)。これが合致した場合には、そのメッセージをアプリケーションプログラムモジュールに受信させてもよいと判断して、メッセージ内のデータ部を該当するアプリケーションプログラムモジュール(複数あればそれら全て)に渡し、この(これら)プログラムモジュールの起動をかける。また管理テーブル611を参照して、起動をかけたプログラムモジュールの属するグループの処理時間タイマーがリセットされている場合には、この処理時間タイマーをスタートさせ(1105)、ステップ1107の処理に移る。合致しない場合はそのメッセージは廃棄し(1106)、ステップ1107の処理に移る。ステップ1101において受信バッファにメッセージがない場合には、ステップ1107の処理に移る。尚、プログラムモジュールへの起動のかけ方、即ちどのプログラムモジュールを実行させるかは、各グループ毎に定める。例えば、複数の起動可能なプログラムモジュールが存在する場合に、一度に全てのプログラムモジュールを起動するか、あるいは、一つのプログラムモジュールの処理が終了するまで待つ一つずつ順番に起動をかける、など各グループでの実行管理の方法は各グループ毎に異なってもよい。

【0037】処理1107では、管理プログラムが管理テーブルによって管理しているアプリケーションプログラムモジュールからの送信データが中間バッファに有るかどうかをチェックする。もしどの中間バッファにも送信すべきデータが存在しない場合はステップ1101に戻る。送信すべきデータが存在する場合には、管理テーブルを検索し、そのアプリケーションプログラムモジュールに対応する欄から、送信情報属性識別子ならびに内容コードを捜し出して、それらをデータに付加する(1108、1111)。

【0038】送信すべきデータに付加した情報から管理テーブル610を見て、そのデータが自処理装置でも必要かどうかを判断する(1112)。必要ない場合にはそのデータを共通伝送媒体に送信し、そのデータを出力したグループの一連の処理が終了したと判断して処理時間タイマーをリセットする(1113)。送信バッファ

内のメッセージは通信ドライバプログラムによって共通伝送媒体に送出される。データがもし自処理装置でも必要な場合には受信バッファにそのデータのコピーを格納し、かつそのデータを共通伝送媒体に送信する。

【0039】本実施例では、通信のプロトコルとして内容コードを用いているが、勿論、広く一般に知られている他の通信方式を用いても構わない。例えば、1対1通信方式にて処理装置の宛先を指定してメッセージを送信し、受け取った処理装置側でメッセージに付加されている情報属性識別子を見てアプリケーションプログラムモジュールに受信させるかどうかを決定する方式であってもよい。

【0040】通信プロトコルとして、内容コードを用いた場合には、各処理装置(プログラムモジュール)は、送信先となる相手を意識することなくメッセージを送信でき、共通伝送媒体から受け取ったメッセージをどう扱うかを受信側の判断に任せることが出来るという特徴を保証することができる。情報が有している質を各処理装置での処理にどのように反映させるかは、システムとして一律に決められるものではなく個々の処理装置あるいは個々のアプリケーションモジュールによって異なる。以上説明した実施例によれば、こうした判断を受信側にて柔軟に、即ち自律的に、行なわせることができる。

【0041】

【発明の効果】本発明によれば、一つの分散システム内に異種の業務が存在したり、その異種業務により扱われる異種の情報が存在する場合に、それらの業務や情報をその属性ごとに管理することが可能となる。これにより、異なる属性の情報の混在がある属性の業務実行に悪影響を与える場合などに、それらの情報を分離したり、関連する属性の情報間で同期をとって業務を実行したりすることが容易に可能となる。

【図面の簡単な説明】

【図1】管理プログラムにより実行される処理の手順を示すフローチャートである。

【図2】本発明が適用される分散計算機システムの一構成例を示すブロック図である。

【図3】プログラムモジュール間で送受信されるメッセージのフォーマット図である。

【図4】内容コードを用いたメッセージのフォーマット図である。

【図5】プログラムモジュール間のデータの流れの一例を示すシステムフロー図である。

【図6】1つの処理装置内に複数のプログラムモジュールグループが存在する場合の処理装置内のデータのながれを示すシステムフロー図である。

【図7】処理装置内のプログラムモジュール間の関係を表すシステムフロー図である。

【図8】各処理装置内のソフトウェアの内部構造を示すブロック図である。

【図9】管理テーブルの構成図である。

【図10】本発明のファイル管理方式を適用した場合の処理装置内でのデータの流れを示すシステムフロー図である。

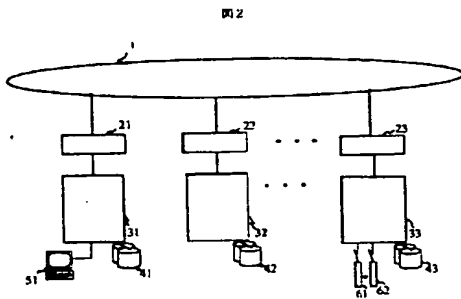
【図11】本発明のファイル管理方式を適用した場合の処理装置間でのデータの流れを示すシステムフロー図である。

【図12】処理装置の構成を示すブロック図である。

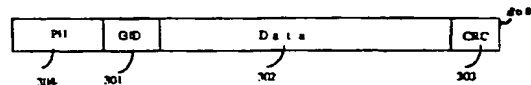
【図13】プログラムモジュール群の処理時間管理の処理手順を示すフローチャートである。

10

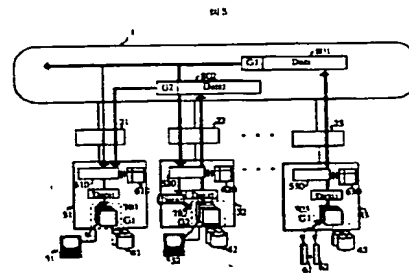
【図2】



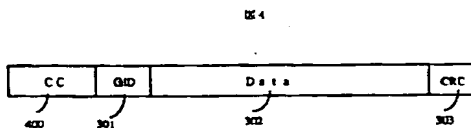
【図3】



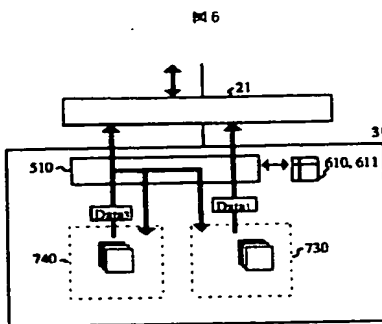
【図5】



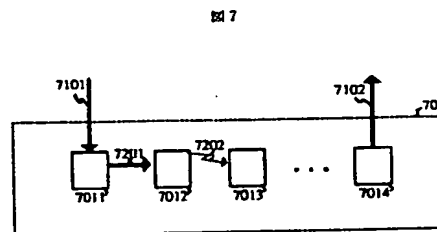
【図4】



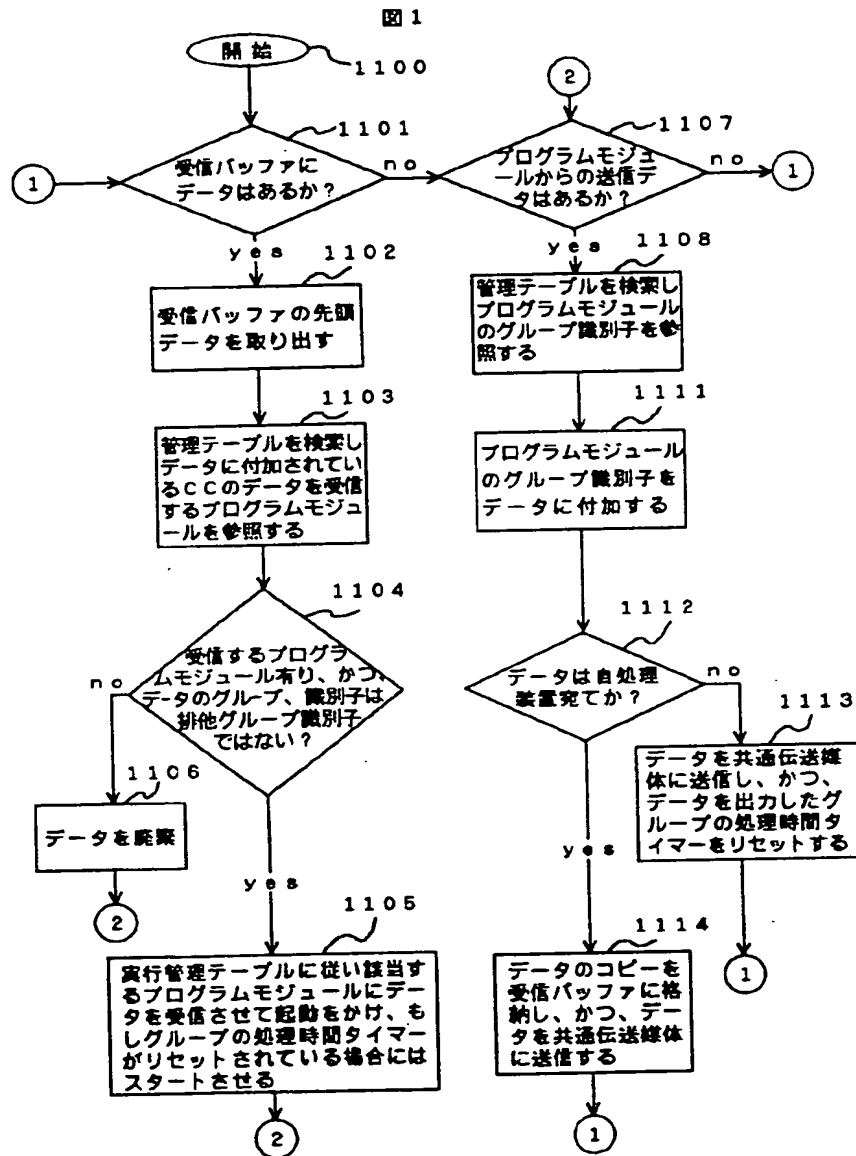
【図6】



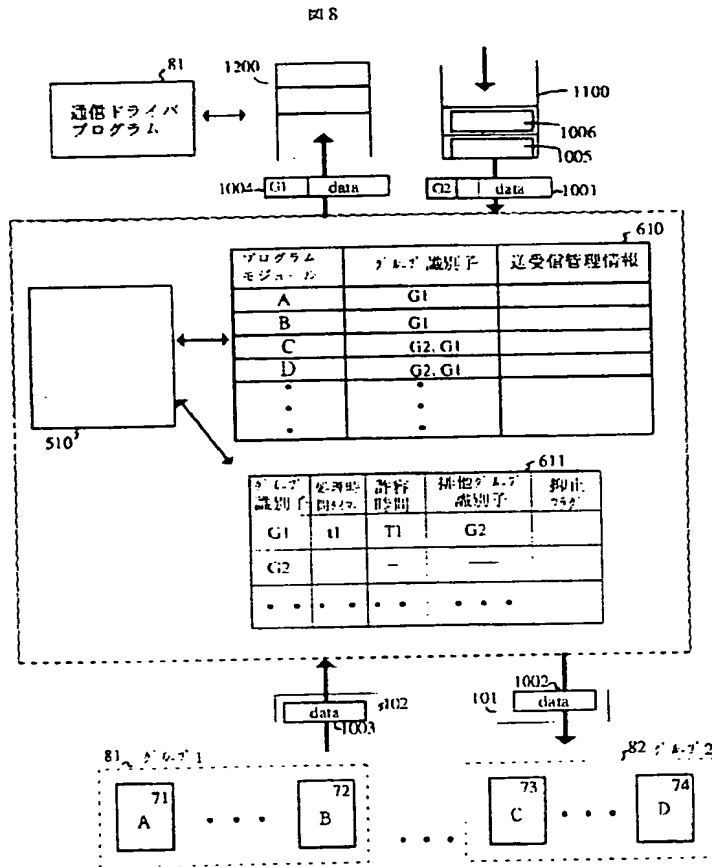
【図7】



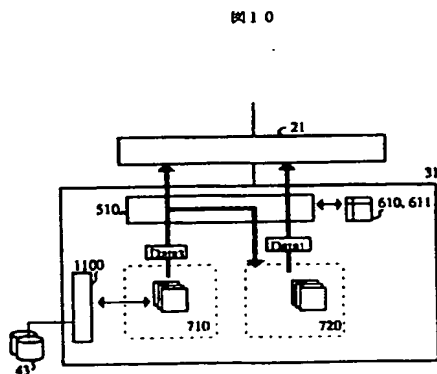
【図1】



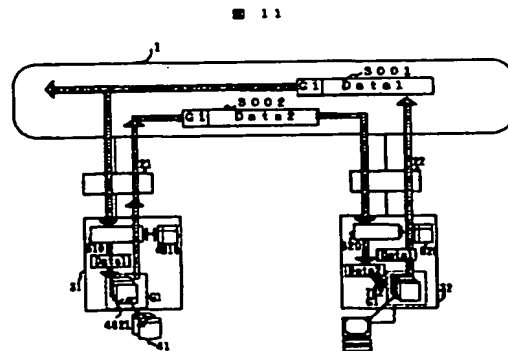
【図8】



【図10】



【図11】



【図9】

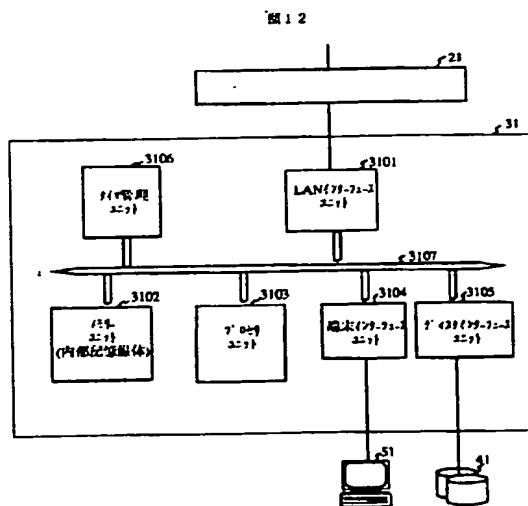
図9

610

| プログラム<br>モジュール | 識別子 | 送信管理情報   |     |          | 受信管理情報   |       |          |     |
|----------------|-----|----------|-----|----------|----------|-------|----------|-----|
|                |     | 送信管理=IT1 | ... | 送信管理=ITN | 受信管理=IT1 | ...   | 受信管理=ITM |     |
| A              | G1  | CC1      | ... |          | CC2      |       | ...      |     |
| B              | G1  | CC3      | ... |          |          |       | ...      |     |
| C              | G2  |          | ... |          | CC4      | data2 | ...      | CC5 |
| ⋮              | ⋮   |          | ⋮   |          |          |       | ⋮        |     |

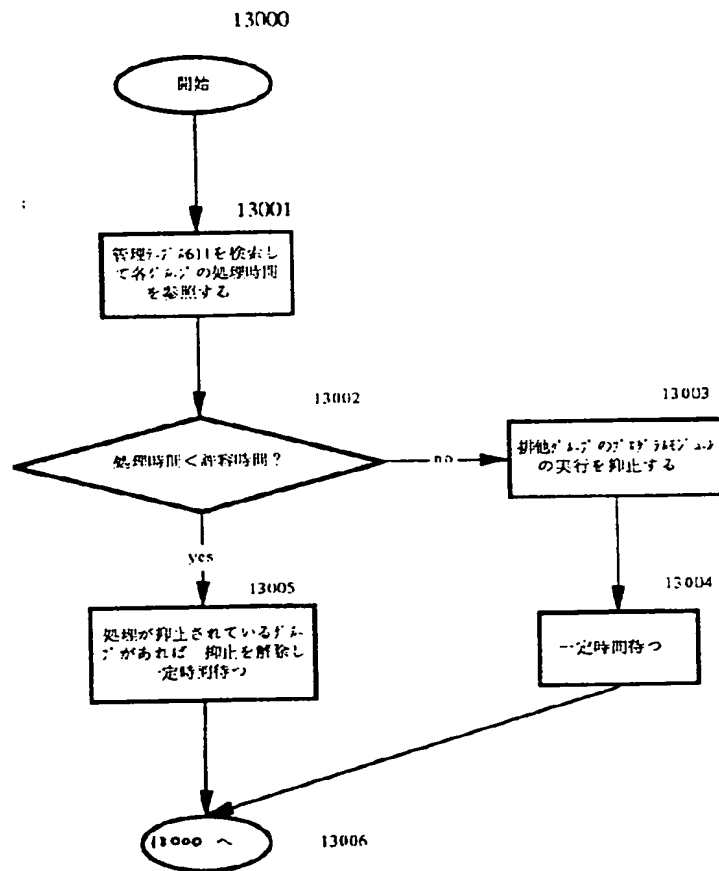
内容データ
内容データ
データ格納領域
内容データ
データ格納領域

【図12】



【図13】

図13



フロントページの続き

(72)発明者 中村 智明  
 茨城県日立市大みか町五丁目2番1号 株  
 式会社日立製作所大みか工場内